Instituto Federal do Sul de Minas Gerais

Projeto e Análise de Algoritmos

(Grafos)

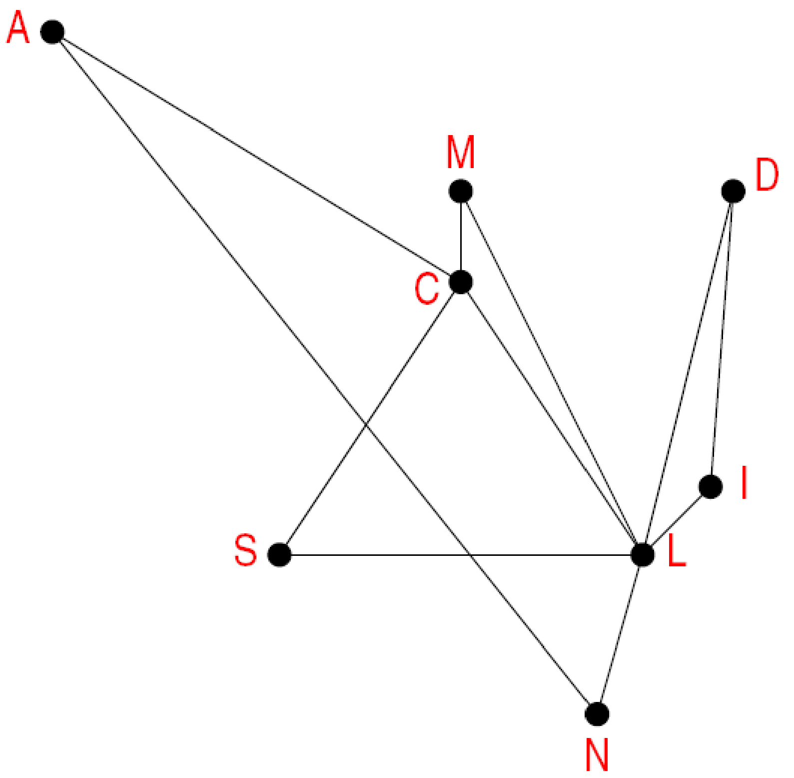
Aula 07 – Árvore Geradora Mínima: Algoritmo Genérico

Prof. Humberto César Brandão de Oliveira
Prof. Douglas Castilho

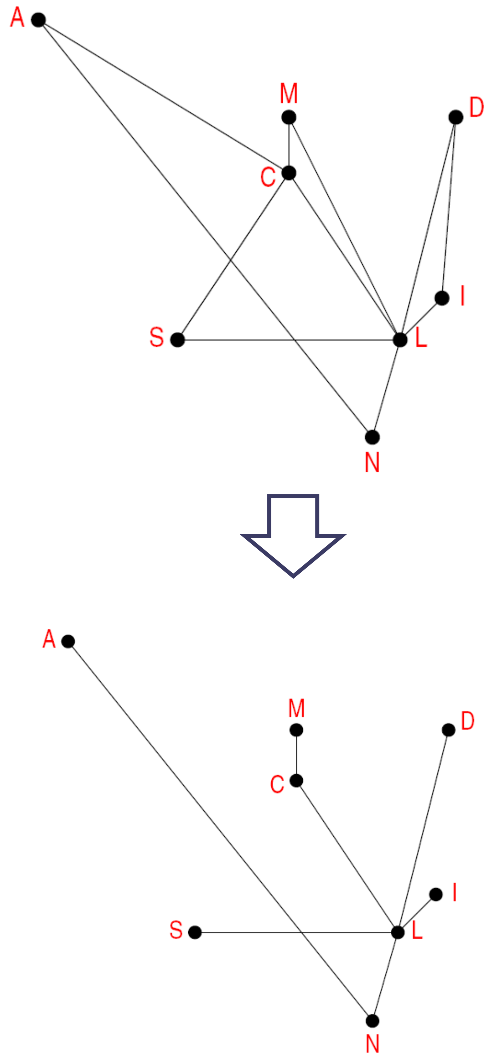
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL DE MINAS GERAIS

Árvore Geradora Mínima

- Suponha que uma companhia aérea recebeu permissão para voar nas rotas da figura...
- Mas por questões de economia, a empresa não irá operar em todas as vias...
- E ao mesmo tempo precisa atender a toda a demanda aérea do país... Afinal, os passageiros podem fazer conexões...



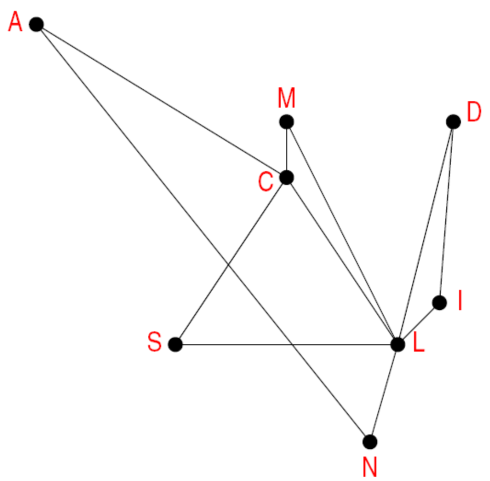
Árvore Geradora Mínima

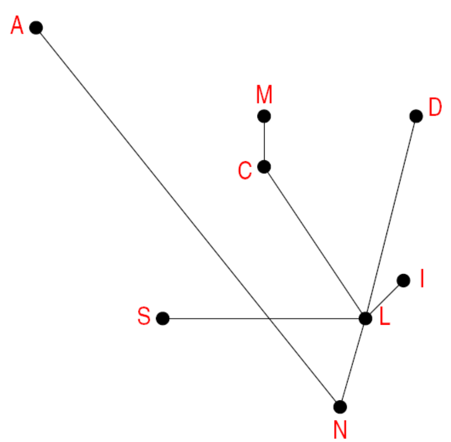


- A segunda figura mostra uma forma de atender toda a demanda, interconectando todas as cidades.
- Este conjunto de rotas é mínimo?
  - Sem considerar os pesos...

Árvore Geradora Mínima

$$G = (V, A)$$





$$G' = (V, X)$$

- Este conjunto de rotas é mínimo?
  - Sim. Qualquer árvore de um grafo de |V| vértices possui |V|-1 arestas.

$$|X| = |V| - 1$$

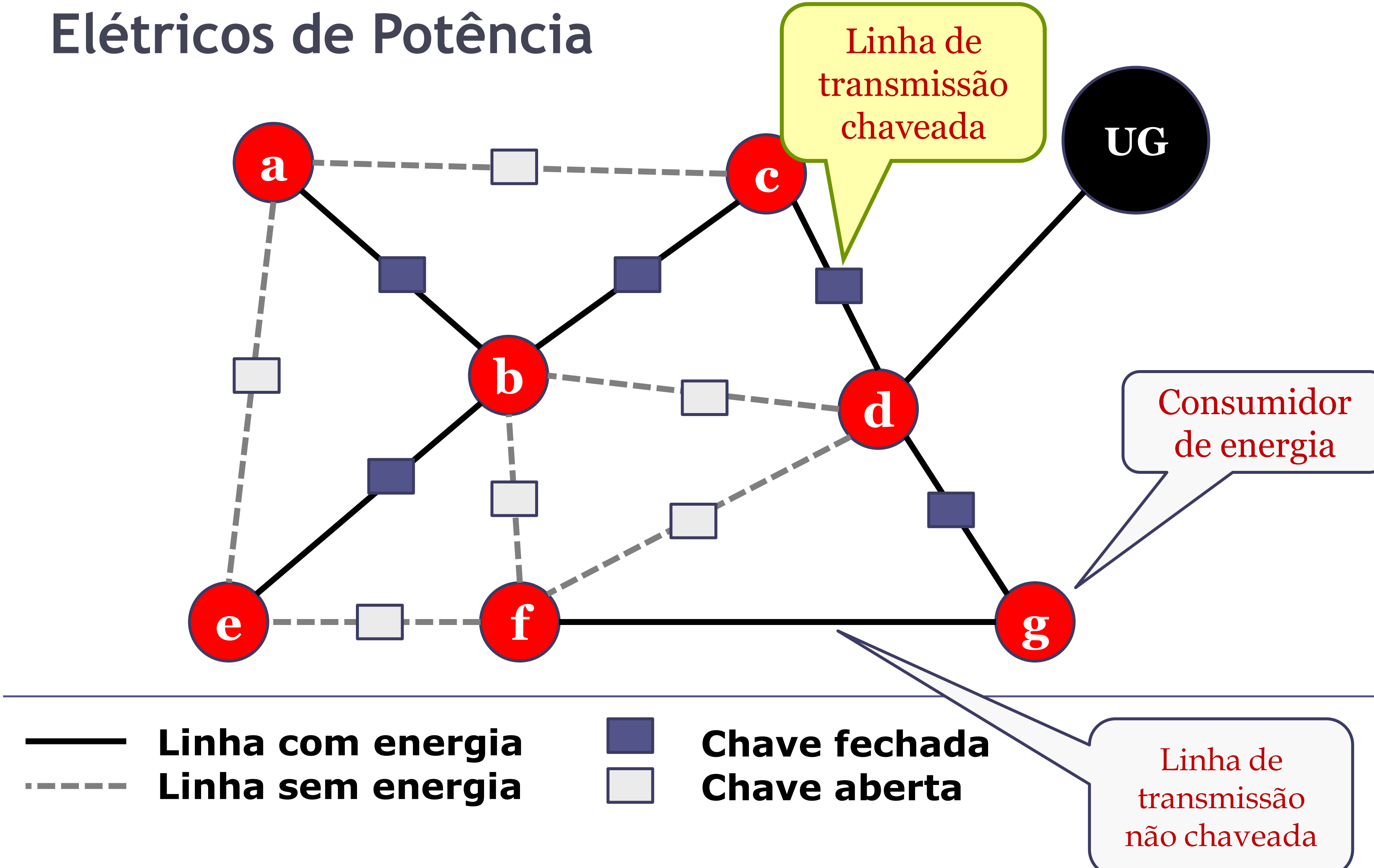
Árvore Geradora Mínima

- História:
  - O cientista checo Otakar Borůvka criou o primeiro algoritmo para encontrar uma árvore geradora mínima em 1926. Ele queria resolver o problema de encontrar uma eficiente cobertura elétrica de Moravia (Região da República Checa);
  - Atualmente este é conhecido como Algoritmo de Borůvka.

Árvore Geradora Mínima

- Aplicações da AGM:
  - Transporte aéreo;
  - Transporte terrestre;
  - Redes de computadores;
    - Exemplo: "Dynamic Minimal Spanning Tree Routing Protocol for Large Wireless Sensor Networks":
      - http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4026014&tag=1
  - Redes elétricas;
  - Circuitos integrados;
  - Etc...

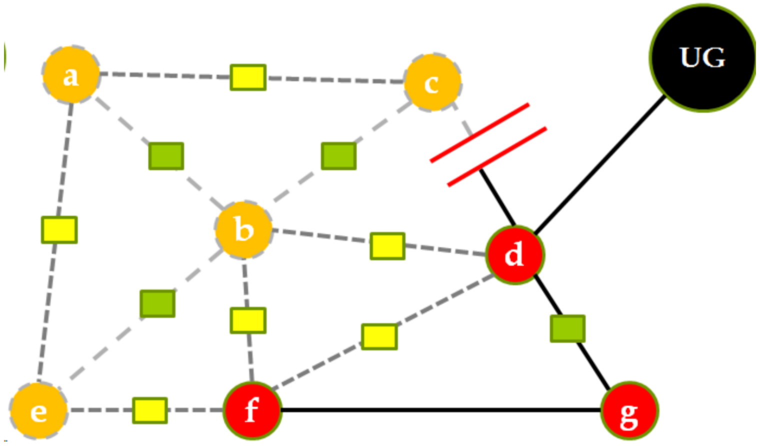
O Problema de Restabelecimento de Sistemas Elétricos de Potência



O Problema de Restabelecimento de Sistemas Elétricos de Potência

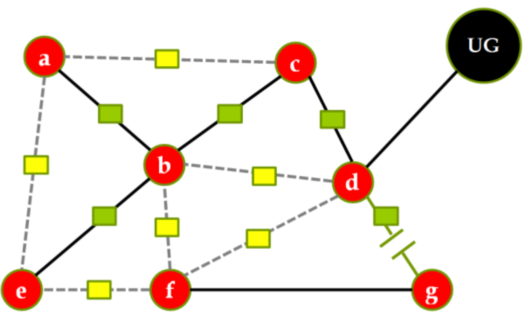
- **Deseja-se 100% de disponibilidade** do SEP;
  - Mas não podemos garantir por falhas como:
    - Catástrofes naturais;
    - Vandalismo;

- A **interrupção** de apenas **uma linha** de transmissão pode causar o **não abastecimento de grande parte dos consumidores**;

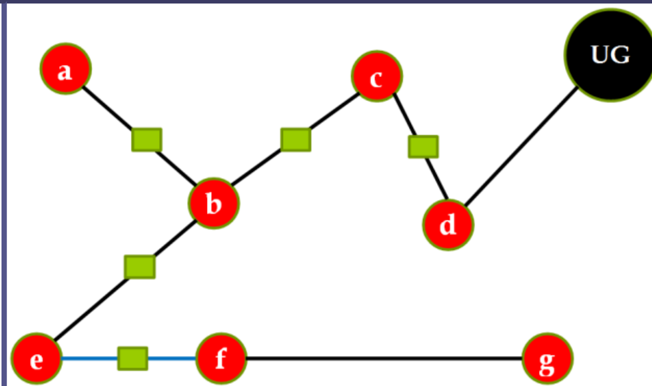


O Problema de Restabelecimento de Sistemas Elétricos de Potência

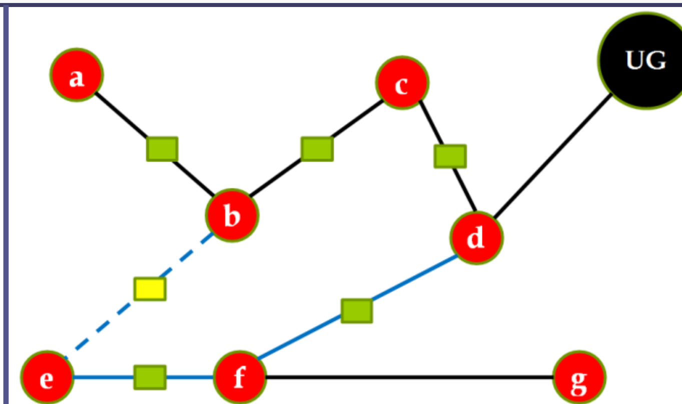
Problema



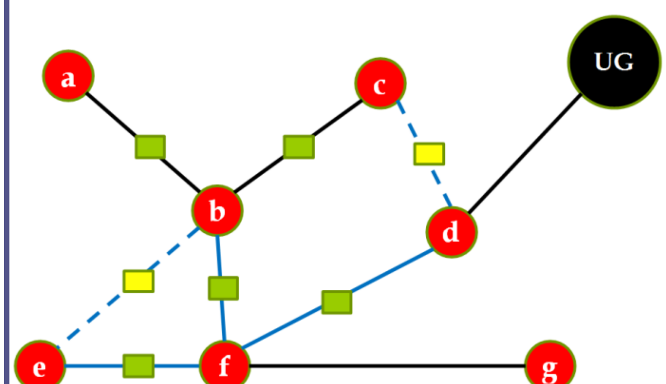
Possíveis soluções para o nosso exemplo



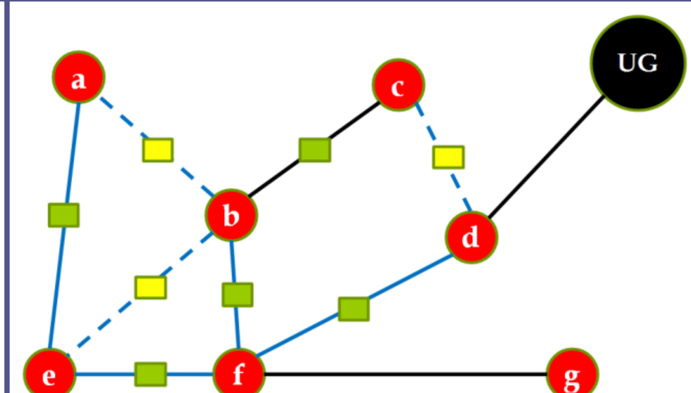
1 alteração



3 alterações



5 alterações



7 alterações

Árvore Geradora Mínima

- Modelando:
  - Para cada aresta *(u,v)* pertencente a A, temos um peso *w(u,v)* especificando o custo para interconectar *(u,v)*. Então, desejamos encontrar um subconjunto acíclico *X* contido ou igual a A, que conecte todos os vértices e cujo peso total seja minimizado.

$$G = (V, A) \Longrightarrow G' = (V, X)$$

$$\min \ w(X) = \sum_{(u,v)\in X} w(u,v)$$

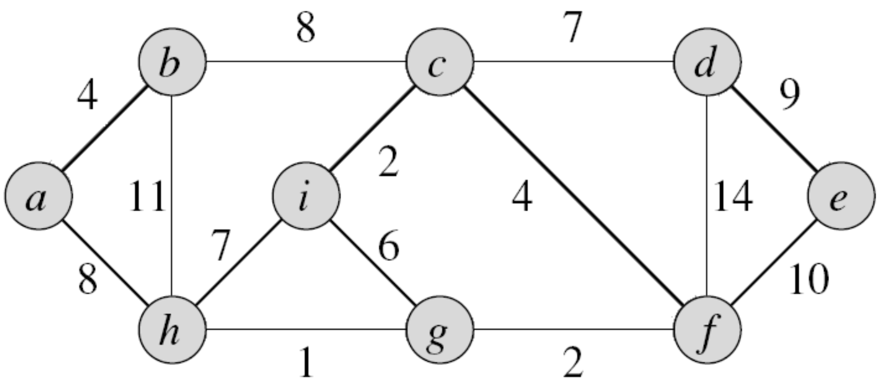
Árvore Geradora Mínima

- Observações:
  - Tendo em vista que *T* é acíclico e conecta todos os vértices, ele deve **formar uma árvore**, que é chamada de:
    - Árvore geradora, ou
    - Árvore espalhada, ou
    - Árvore de extensão;
  - O problema de determinar a **árvore de menor custo** é conhecido como:
    - Problema da Árvore Geradora Mínima, ou
    - Problema da Árvore Espalhada Mínima.
    - Problema da Árvore de Extensão Mínima.

Árvore Geradora Mínima

- Exemplo:

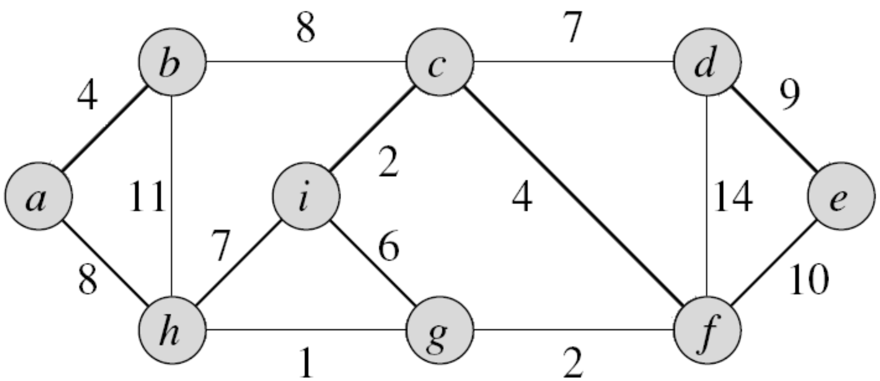
$$G = (V, A)$$



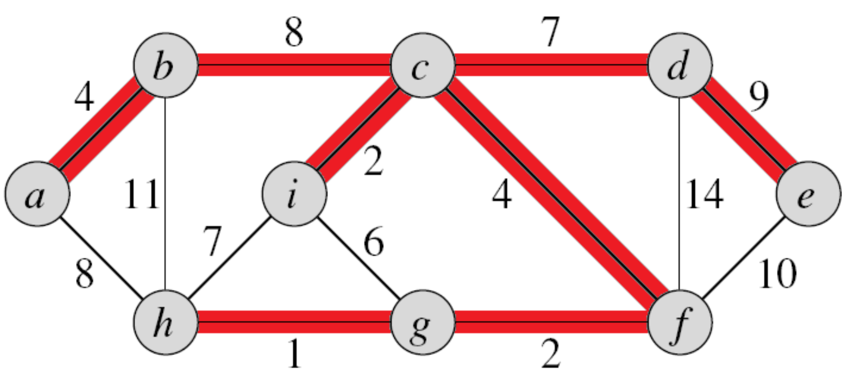
Árvore Geradora Mínima

- Exemplo:

$$G = (V, A)$$



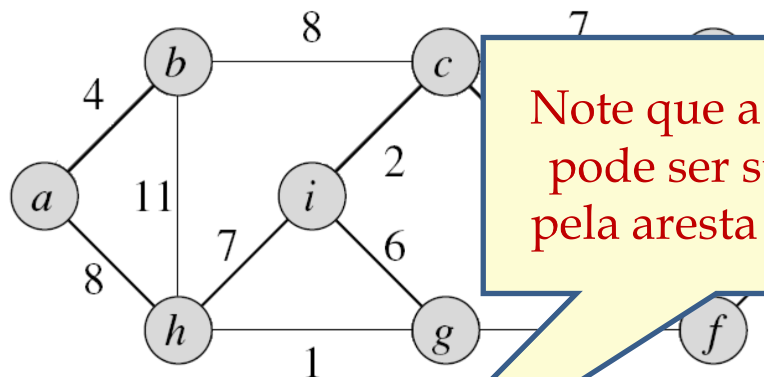
$$G' = (V, X)$$



Árvore Geradora Mínima

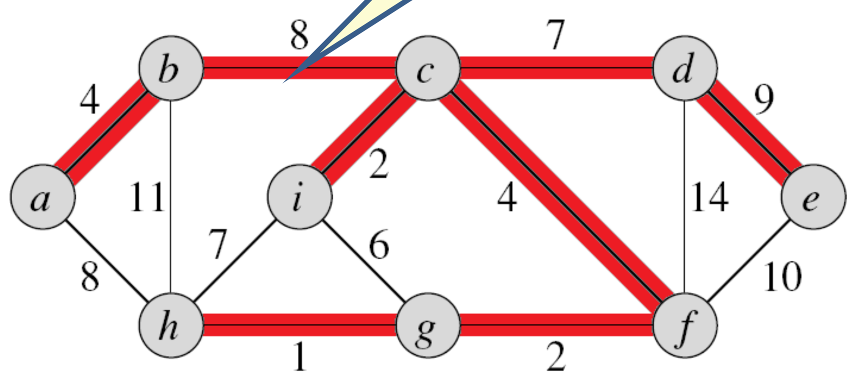
- Exemplo:

$$G = (V, A)$$



Note que a aresta *(b,c)* pode ser substituída pela aresta *(a,h)* em *A′*

$$G' = (V, X)$$



Árvore Geradora Mínima

- Existem dois algoritmos clássicos na literatura para resolver o problema da AGM:
  - Algoritmo de Prim;
  - Algoritmo de Kruskal;

- Ambos são considerados algoritmos gulosos.

- A estratégia gulosa defende que a menor escolha a cada passo deve ser feita, mesmo que tal escolha não nos leve a uma solução ótima ao final da execução.
  - MELHOR ESCOLHA IMEDIATA...

Árvore Geradora Mínima

- Grafo não orientado: $G = (V, A)$
- Peso nas arestas: $w : A \to \Re$
- Antes de cada iteração, X representa o subconjunto de arestas de alguma árvore geradora mínima;
- A cada iteração uma aresta *(u,v)* é adicionada ao conjunto X.
- Problema: Como definir qual aresta do grafo original pode fazer parte de alguma AGM?

$$X \leftarrow X \cup \{(u, v)\}$$

Árvore Geradora Mínima

- Algoritmo Genérico para AGM:

$$AGM\_GENERICA(\ G(V, A),\ w\ )$$

$$X \leftarrow \{\ \ \}$$

$$enquanto\ |X| \neq (|V| - 1)\ faça$$

$$encontrar\ \ uma\ aresta\ (u, v)\ segura\ \ para\ X$$

$$X \leftarrow X \cup \{(u, v)\}$$

$$fim\ \ enquanto$$

$$retorna\ \ X$$

$$fim.$$

Árvore Geradora Mínima

- Algoritmo Genérico para AGM:

$$AGM\_GENERICA(\ G(V,A),\ w\ )$$
$$X \leftarrow \{\ \ \}$$
$$enquanto\ |X| \neq (|V|-1)\ faça$$
$$encontrar\ \ uma\ aresta\ (u,v)\ segura\ \ para\ \ X$$
$$X \leftarrow X \cup \{(u,v)\}$$
$$fim\ \ enquanto$$
$$retorna\ \ X$$
$$fim.$$

É obvio que o ponto chave é a localização da aresta que pode fazer parte de alguma AGM.

Árvore Geradora Mínima

- Antes de fornecermos uma forma de identificar uma aresta segura para a AGM, precisamos de um conceito na área de grafos: corte

- Corte é uma partição do conjunto de vértices

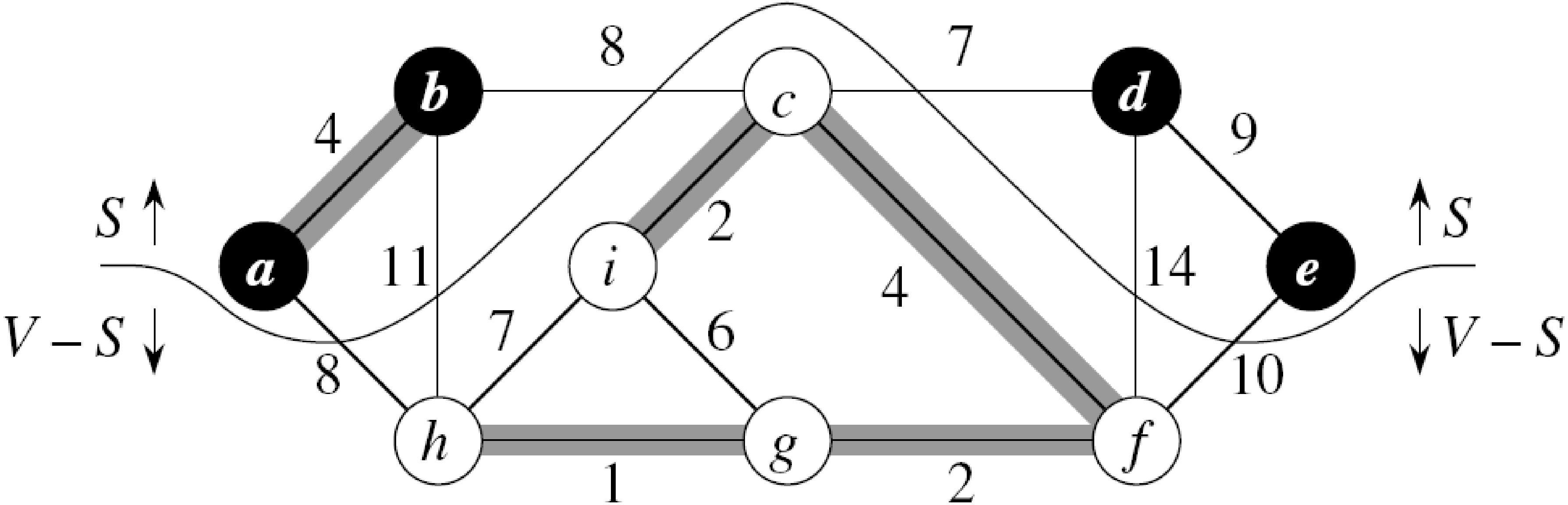
- Corte:

$$G = (V, A)$$

$$(S, V - S)$$

Árvore Geradora Mínima

- Primeira maneira de visualizar um corte:



$$S = \{a, b, d, e\}$$

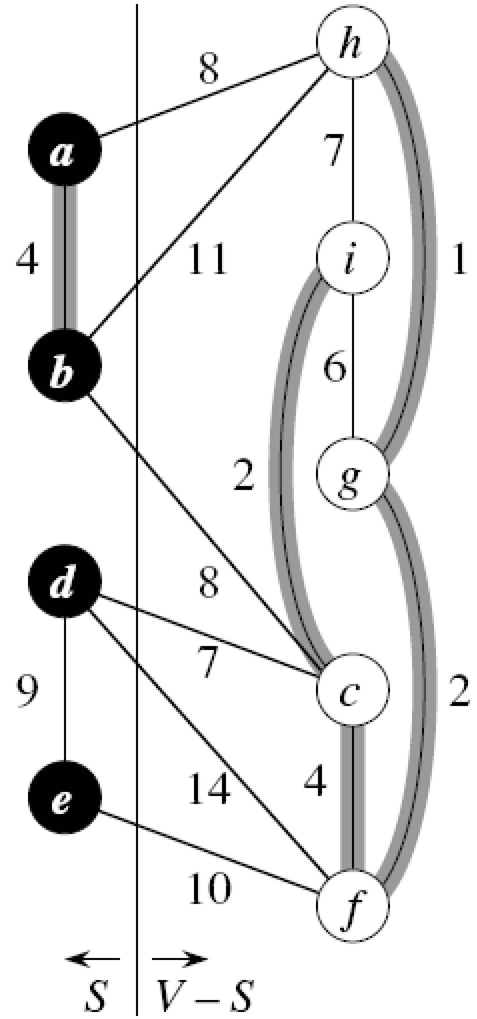
$$V - S = \{h, i, c, g, f\}$$

Árvore Geradora Mínima

- Segunda maneira de visualizar o mesmo corte:

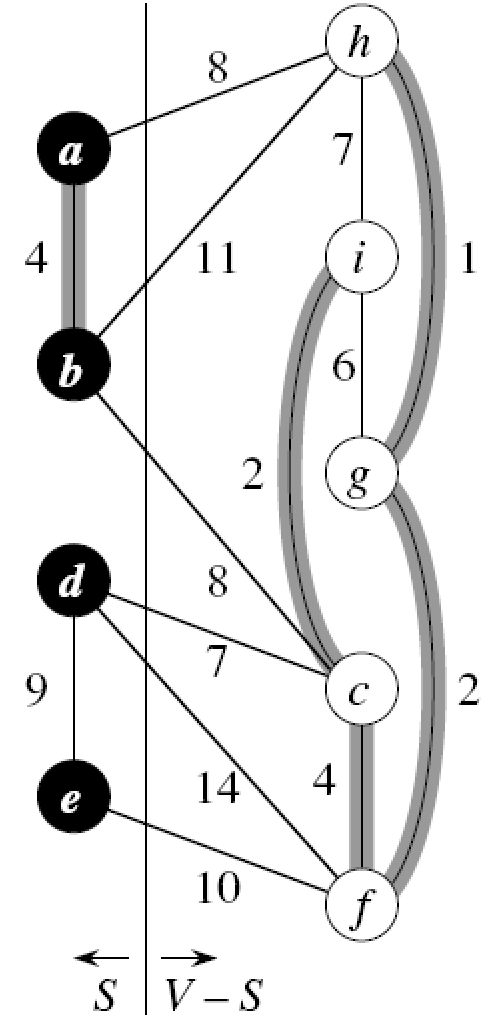
$$S = \{a, b, d, e\}$$

$$V - S = \{h, i, c, g, f\}$$



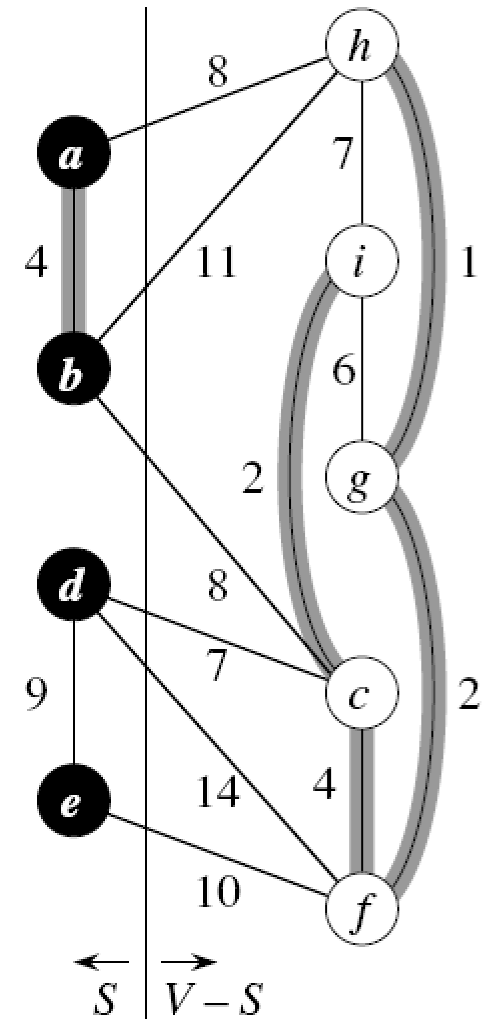
Árvore Geradora Mínima

- Definição 1:
  - ARESTA QUE CRUZA O CORTE
  - Dizemos que a aresta (u,v) cruza o corte se um dos seus pontos está em S, e o outro está em V-S;



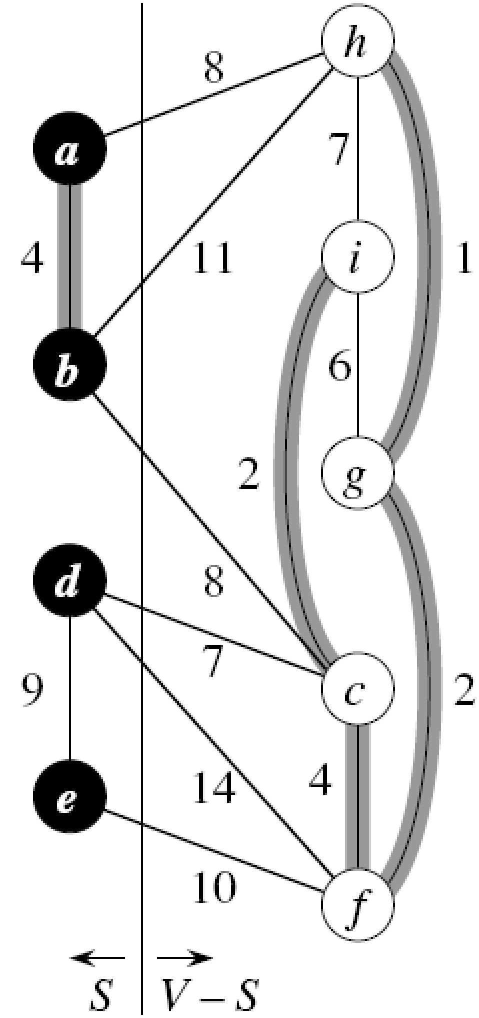
Árvore Geradora Mínima

- Definição 2:
  - CORTE RESPEITA X
  - Dizemos que **um corte respeita o conjunto X** se nenhuma aresta de X cruza o corte.



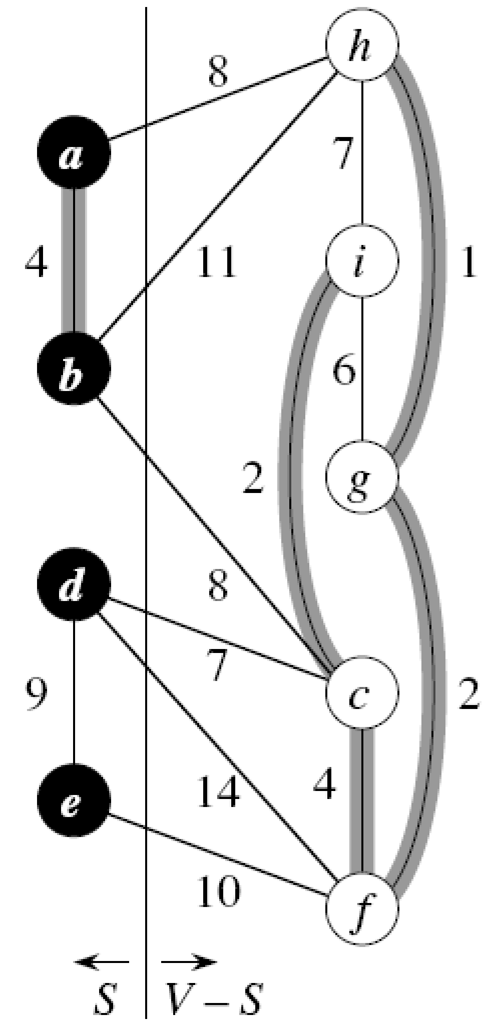
Árvore Geradora Mínima

- Definição 3:
  - ARESTA LEVE
  - Dizemos que uma aresta é **uma aresta leve** cruzando o corte se o seu peso é o menor, se comparado as outras arestas que cruzam o corte.



Árvore Geradora Mínima

- Arestas seguras:
  - **Teorema:** Seja *(S, V-S)* qualquer corte de *G* que respeita *X* e seja *(u,v)* uma aresta leve cruzando *(S,V-S)*, então a aresta *(u,v)* é segura para *X*.



Árvore Geradora Mínima

- <u>Voltando ao algoritmo:</u>

$AGM\_GENERICA(\ G(V,A),\ w\ )$

$\quad X \leftarrow \{\ \ \}$

$\quad enquanto\ |X| \leq |V|-1\ \ faça$

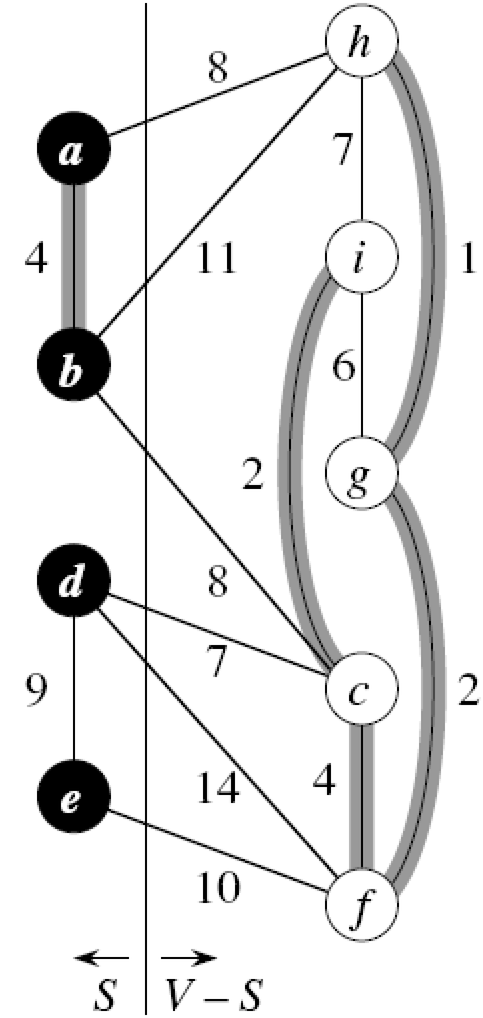
$\quad\quad encontrar\ \ uma\ aresta\ (u,v)\ segura\ \ para\ X$

$\quad\quad X \leftarrow X \cup \{(u,v)\}$

$\quad fim\ \ enquanto$

$\quad retorna\ \ X$

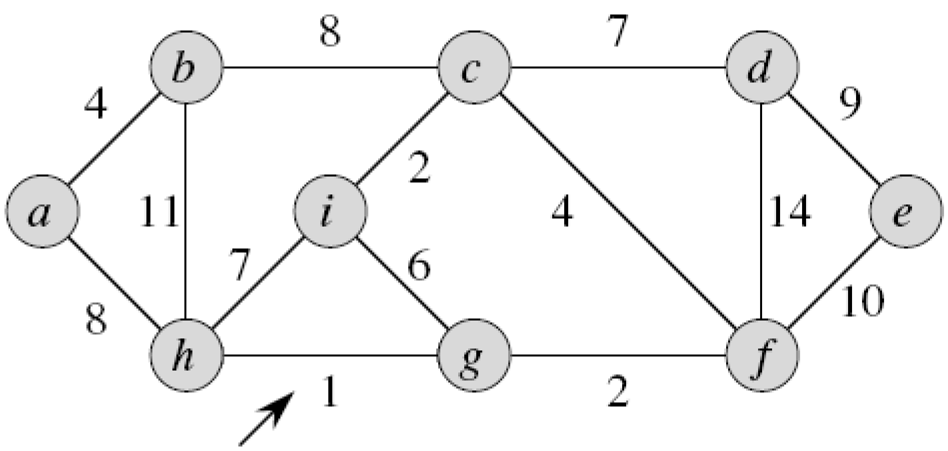
$fim.$



Árvore Geradora Mínima

- Questão 1:
  - Seja *(u,v)* uma aresta de peso mínimo em um grafo *G*:

    - Ela pode <u>não</u> pertencer a AGM?



Árvore Geradora Mínima

- Dois algoritmos clássicos para a AGM:
  - Kruskal;
  - Prim;

- Primeiro algoritmo:
  - Boruvka.

Árvore Geradora Mínima

- Prim:
  - *Gera uma árvore única;*
  - *Ao longo do algoritmo, o conjunto X sempre é uma árvore.*

- Kruskal:
  - *Gera uma floresta, antes de gerar a AGM;*
  - *Existe garantia de ser uma árvore apenas depois da última iteração.*

Kruskal

- Na aula de hoje vamos estudar o algoritmo de Kruskal:
  - Criado por Joseph Bernard Kruskal, Jr.
  - Nascido em 1928.
  - Terminou seu PhD na Universidade de Princeton em 1956

Árvore Geradora Mínima

- <u>Arestas seguras:</u>
  - <u>Prim:</u>
    - *A aresta segura é sempre a aresta de peso mínimo que conecta a árvore a um vértice não presente no conjunto X.*
  - <u>Kruskal:</u>
    - *A aresta segura é sempre uma aresta de peso mínimo no grafo que conecta dois componentes distintos (duas árvores distintas na floresta).*

Kruskal

- Ponto Chave:
  - Ele encontra uma **aresta segura** para adicionar à floresta encontrando, **de todas as arestas que conectam duas árvores quaisquer, uma aresta de peso mínimo**;
    - Se você reparar, **o corte acontece neste ponto**... Mas para isso, é efetuada uma adaptação no grafo original
- Kruskal é considerado um algoritmo guloso, porque em cada passo ele adiciona à floresta uma aresta de peso mínimo (daquelas que ainda podem ser adicionadas).
  - Ou seja, faz uma avaliação dentre todas as possibilidades que possui;

Kruskal

$$AGM\_Kruskal(G(V,A), w)$$

$$X \leftarrow \{\ \}$$

$$para\ cada\ vértice\ v \in V\ \ faça$$

$$criarConjunto(v)$$

$$fim\ \ para$$

$$A' \leftarrow ordenar\ as\ arestas\ de\ A\ por\ peso\ crescente$$

$$para\ cada\ aresta\ (u,v) \in A'\ faça$$

$$se\ conjuntoDe(u) \neq conjuntoDe(v)\ então$$

$$X \leftarrow X \cup \{(u,v)\}$$

$$aplicarUnião(u,v)$$

$$fim\ \ se$$

$$fim\ \ para$$

$$retorne\ \ X$$

$$fim.$$

Kruskal

$AGM\_Kruskal(G(V, A), w)$

$X \leftarrow \{\ \}$

$para\ cada\ vértice\ v \in V\ \ faça$

$criarConjunto(v)$

$fim\ \ para$

$A' \leftarrow ordenar\ as\ arestas\ de\ A\ por\ peso\ crescente$

$para\ cada\ aresta\ (u,v) \in A'\ faça$

$se\ conjuntoDe(u) \neq conjuntoDe(v)\ então$

$X \leftarrow X \cup \{(u,v)\}$

$aplicarUnião(u,v)$

$fim\ \ se$

$fim\ \ para$

$retorne\ \ X$

$fim.$

A princípio, o conjunto que guarda as arestas da AGM é vazio.

Kruskal

$AGM\_Kruskal(G(V,A), w)$

$X \leftarrow \{\ \}$

$para\ cada\ vértice\ v \in V\ \ faça$

$criarConjunto(v)$

$fim\ \ para$

$A' \leftarrow ordenar\ as\ arestas\ de\ A\ por\ peso\ crescente$

$para\ cada\ aresta\ (u,v) \in A'\ faça$

$se\ conjuntoDe(u) \neq conjuntoDe(v)\ então$

$X \leftarrow X \cup \{(u,v)\}$

$aplicarUnião(u,v)$

$fim\ \ se$

$fim\ \ para$

$retorne\ \ X$

$fim.$

|V| árvores são criadas.

Kruskal

$AGM\_Kruskal(G(V, A), w)$

$\quad X \leftarrow \{\ \}$

$\quad para\ cada\ vértice\ v \in V\ \ faça$

$\qquad criarConjunto(v)$

$\quad fim\ \ para$

$\quad A' \leftarrow ordenar\ as\ arestas\ de\ A\ por\ peso\ crescente$

$\quad para\ cada\ aresta\ (u, v) \in A'\ faça$

$\qquad se\ conjuntoDe(u) \neq conjuntoDe(v)\ então$

$\qquad\quad X \leftarrow X \cup \{(u, v)\}$

$\qquad\quad aplicarUnião(u, v)$

$\qquad fim\ \ se$

$\quad fim\ \ para$

$\quad retorne\ \ X$

$fim.$

O conjunto de arestas é ordenado em função dos pesos. Condição necessária para a criação da AGM através de Kruskal

Kruskal

$AGM\_Kruskal(G(V,A),\ w)$

$X \leftarrow \{\ \}$

$para\ cada\ vértice\ v \in V\ \ faça$

$criarConjunto(v)$

$fim\ \ para$

$A' \leftarrow ordenar\ \ as\ \ arestas\ \ de\ \ A\ por\ \ peso\ \ crescente$

$para\ \ cada\ \ aresta\ (u,v) \in A'\ \ faça$

$se\ conjuntoDe(u) \neq conjuntoDe(v)\ \ então$

$X \leftarrow X \cup \{(u,v)\}$

$aplicarUnião(u,v)$

$fim\ \ se$

$fim\ \ para$

$retorne\ \ X$

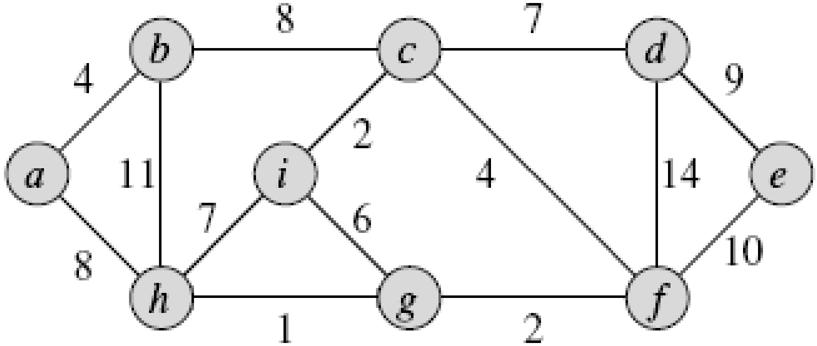
$fim.$

Para cada aresta do vetor ordenado

Se u e v são de árvores distintas, a aresta (u,v) é adicionada ao conjunto X e é aplicada uma união das árvores de u e v.

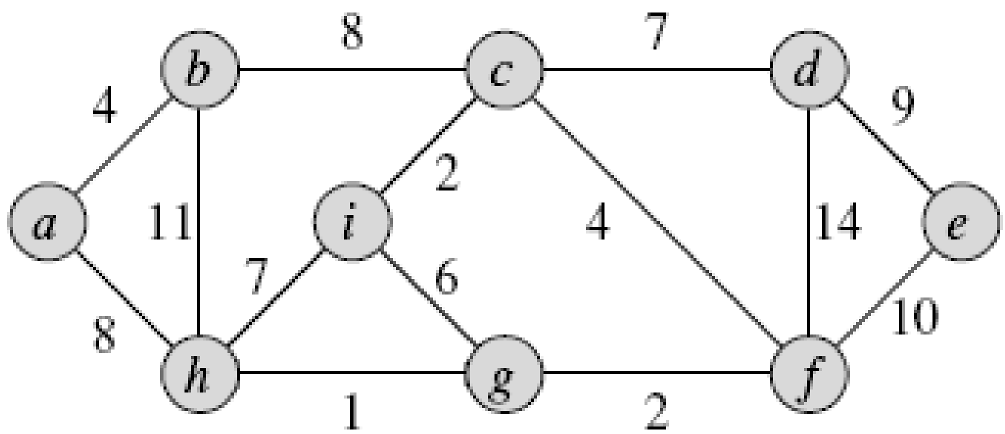
AGM utilizando Kruskal

- Considerando o grafo a seguir... Vamos criar passo-a-passo a AGM utilizando Kruskal...



AGM utilizando Kruskal

- 1º passo: criar um(a) conjunto/árvore para cada vértice.

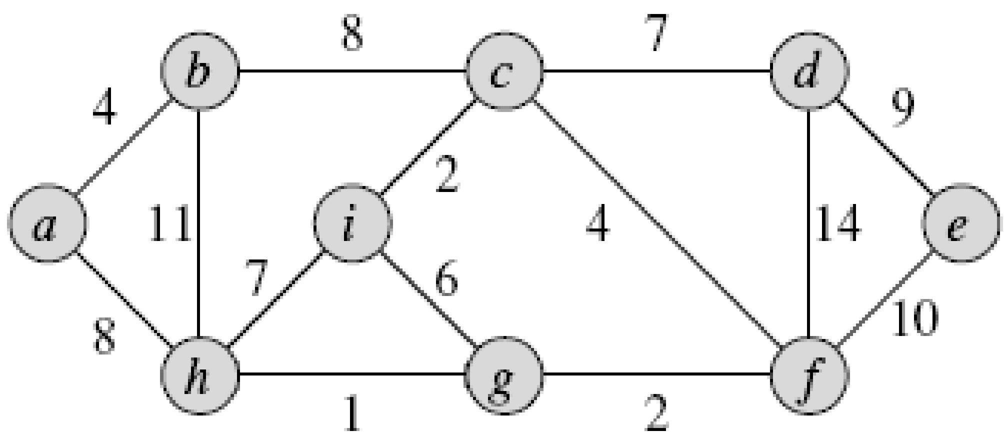


$$\{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\},\{g\},\{h\},\{i\}\}$$

AGM utilizando Kruskal

- 2º passo: ordenar as arestas do conjunto A.

$$\{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\},\{g\},\{h\},\{i\}\}$$



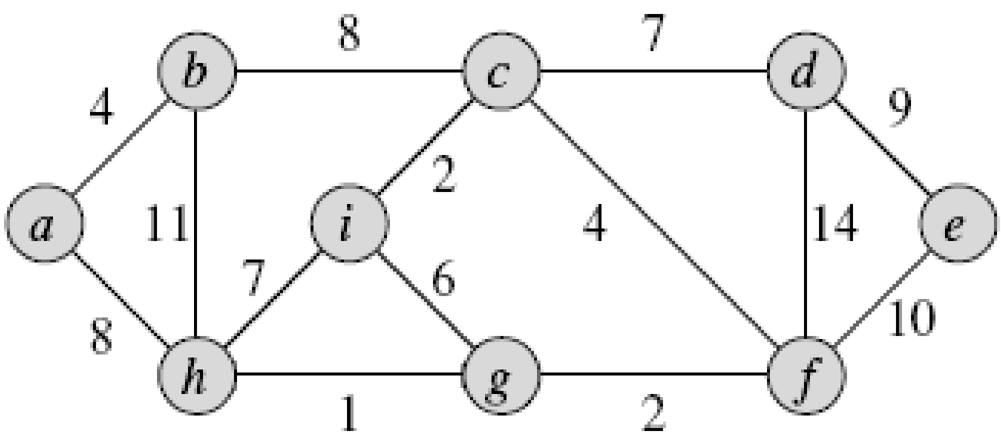
A' $(g,h);(c,i);(f,g);(a,b);(c,f);(g,i);(c,d);(h,i);$
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- <u>3º passo:</u> para cada aresta ordenada, faça...

$$\{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\},\{g\},\{h\},\{i\}\}$$

g e h pertencem a mesma árvore na floresta?



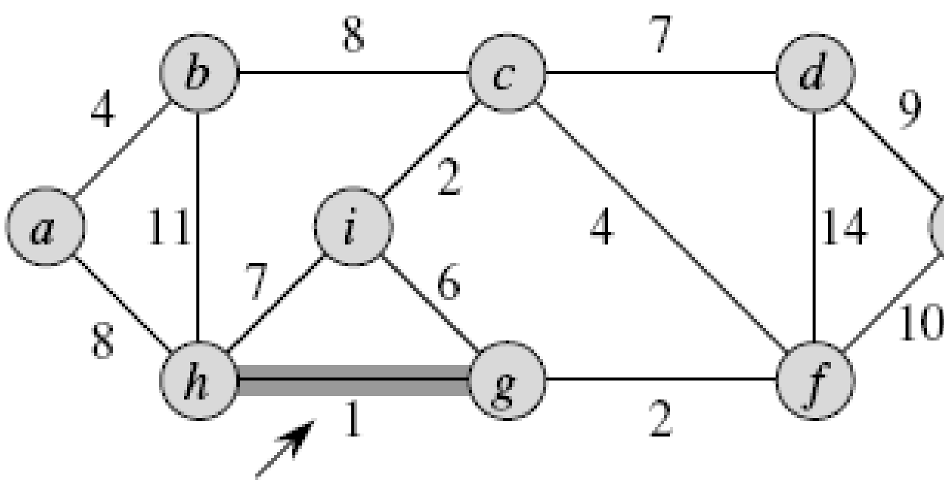
A' $\boxed{(g,h);}(c,i);(f,g);(a,b);(c,f);(g,i);(c,d);(h,i);$
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\},\boxed{\{g,h\}},\{i\}\}$$

g e h pertencem a mesma árvore na floresta?



Não... Então...
União das árvores de g e h e adição da aresta na AGM

A' $\boxed{(g,h);}(c,i);(f,g);(a,b);(c,f);(g,i);(c,d);(h,i);$

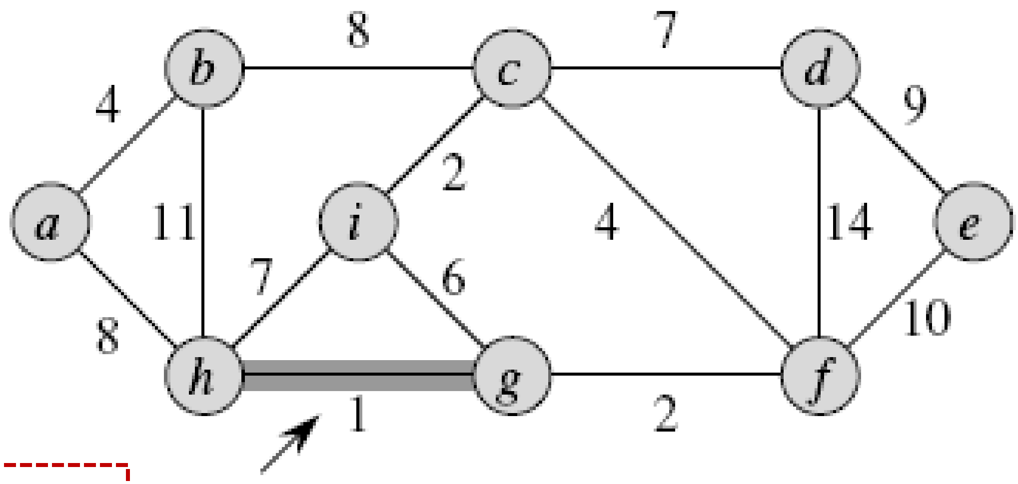
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\},\{g,h\},\{i\}\}$$

c e *i* pertencem a mesma árvore na floresta?



A' $\quad (g,h);\ \boxed{(c,i);}\ (f,g);(a,b);(c,f);(g,i);(c,d);(h,i);$

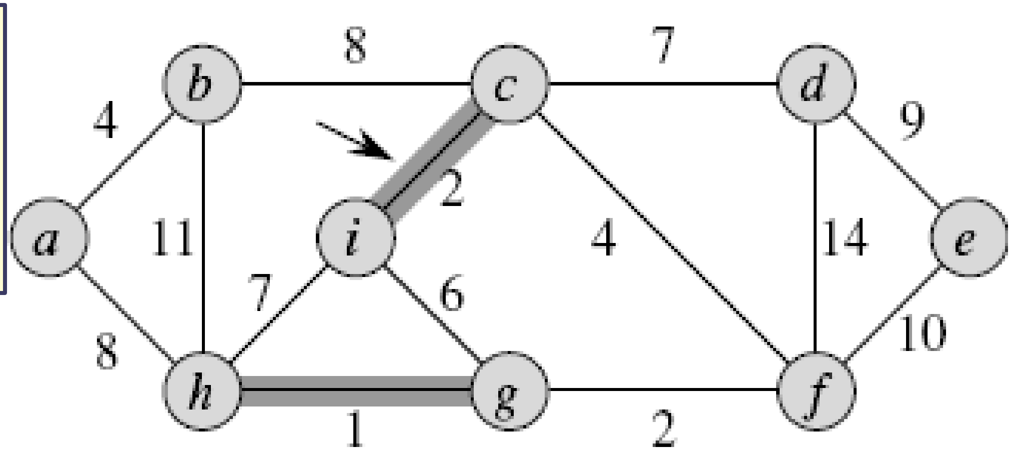
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{a\},\{b\},\boxed{\{c,i\},}\{d\},\{e\},\{f\},\{g,h\}\}$$

Não... Então...
União das árvores
de *c* e *i* e adição da
aresta na AGM



A' $(g,h);\boxed{(c,i);}(f,g);(a,b);(c,f);(g,i);(c,d);(h,i);$

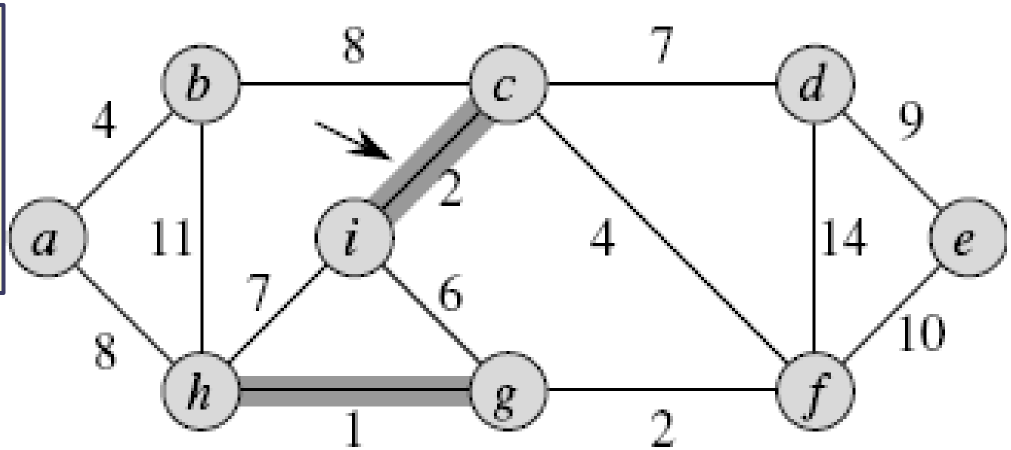
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{a\},\{b\},\{c,i\},\{d\},\{e\},\{f\},\{g,h\}\}$$

f e *g* pertencem a mesma árvore na floresta?



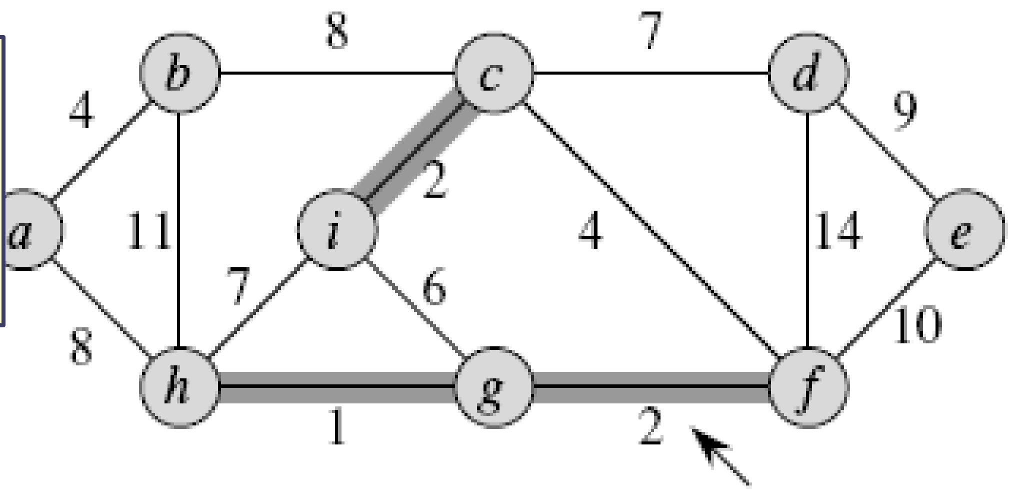
A' $(g,h); (c,i); \boxed{(f,g);} (a,b); (c,f); (g,i); (c,d); (h,i);$
$(a,h); (b,c); (d,e); (e,f); (b,h); (d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{a\},\{b\},\{c,i\},\{d\},\{e\},\{f,g,h\}\}$$

Não... Então... União das árvores de *f* e *g* e adição da aresta na AGM



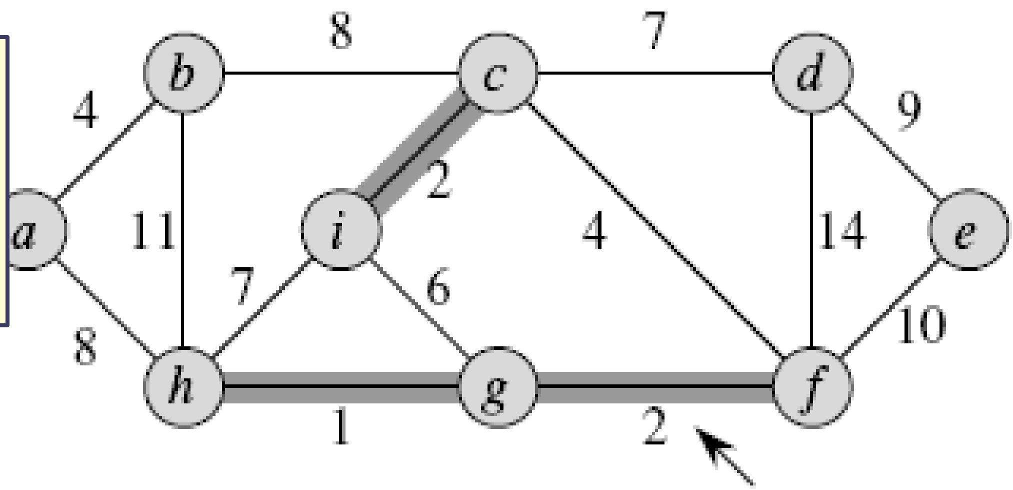
A' $(g,h);(c,i);(f,g);(a,b);(c,f);(g,i);(c,d);(h,i);$
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{a\},\{b\},\{c,i\},\{d\},\{e\},\{f,g,h\}\}$$

a e *b* pertencem a mesma árvore na floresta?



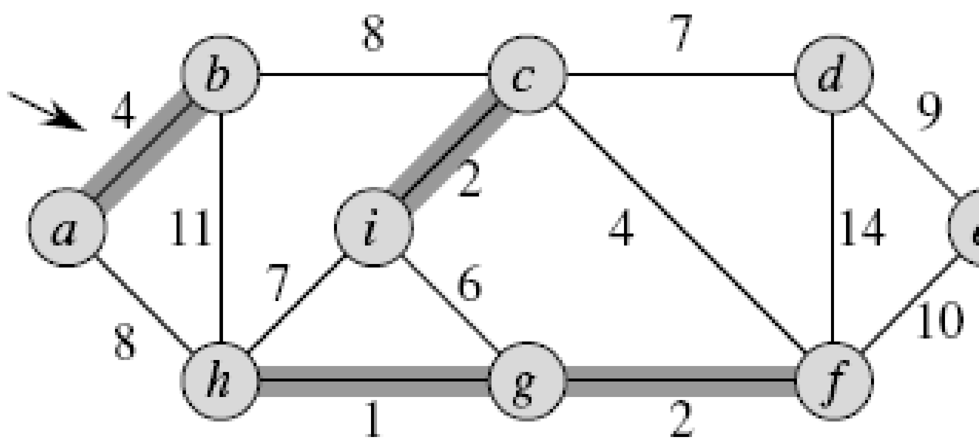
A' $(g,h);(c,i);(f,g);\boxed{(a,b);}(c,f);(g,i);(c,d);(h,i);$

$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\left\{\boxed{\{a,b\}},\{c,i\},\{d\},\{e\},\{f,g,h\}\right\}$$



Não... Então... União das árvores de *a* e *b* e adição da aresta na AGM

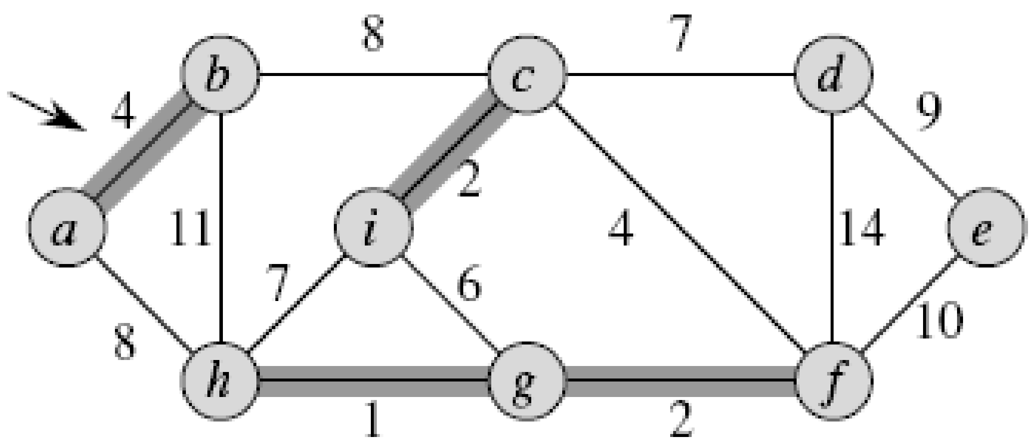
A' $(g,h);\ (c,i);(f,g);\boxed{(a,b);}(c,f);(g,i);(c,d);(h,i);$
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

c e *f* pertencem a mesma árvore na floresta?

- 3º passo: para cada aresta ordenada, faç

$$\{\{a,b\},\{c,i\},\{d\},\{e\},\{f,g,h\}\}$$



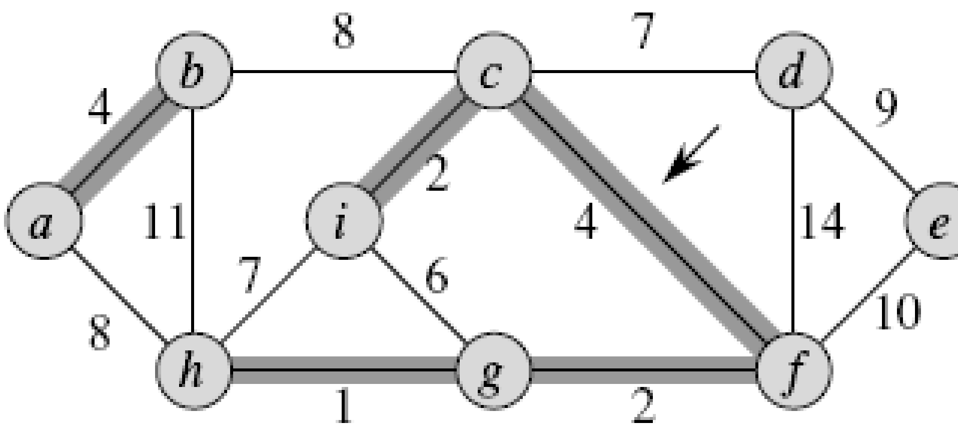
A' $(g,h);(c,i);(f,g);(a,b);\boxed{(c,f);}(g,i);(c,d);(h,i);$

$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{a,b\},\{d\},\{e\},\boxed{\{c,f,g,h,i\}}\}$$



Não... Então...
União das árvores de c e f e adição da aresta na AGM

A' $(g,h);(c,i);(f,g);(a,b);\boxed{(c,f)};(g,i);(c,d);(h,i);$

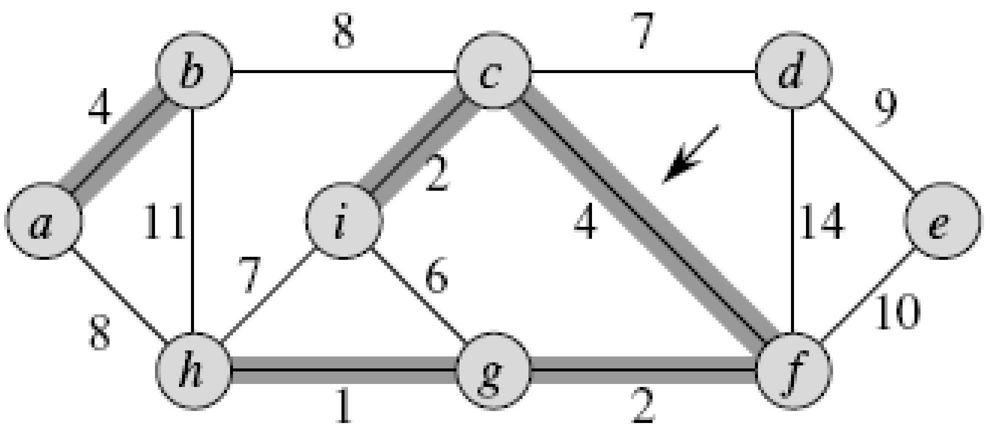
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

g e i pertencem a mesma árvore na floresta?

- 3º passo: para cada aresta ordenada, faç

$$\{\{a,b\},\{d\},\{e\},\{c,f,g,h,i\}\}$$



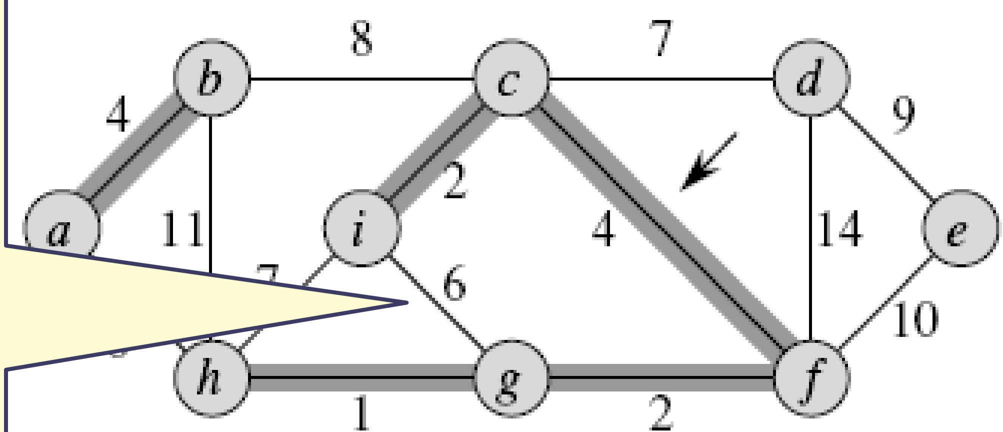
A' $(g,h);(c,i);(f,g);(a,b);(c,f);\boxed{(g,i);}(c,d);(h,i);$
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{a,b\},\{d\},\{e\},\boxed{\{c,f,g,h,i\}}\}$$

(g,i) fecha um ciclo. Isso é identificado porque *g* e *i* pertencem a mesma árvore na estrutura auxiliar 'floresta'.



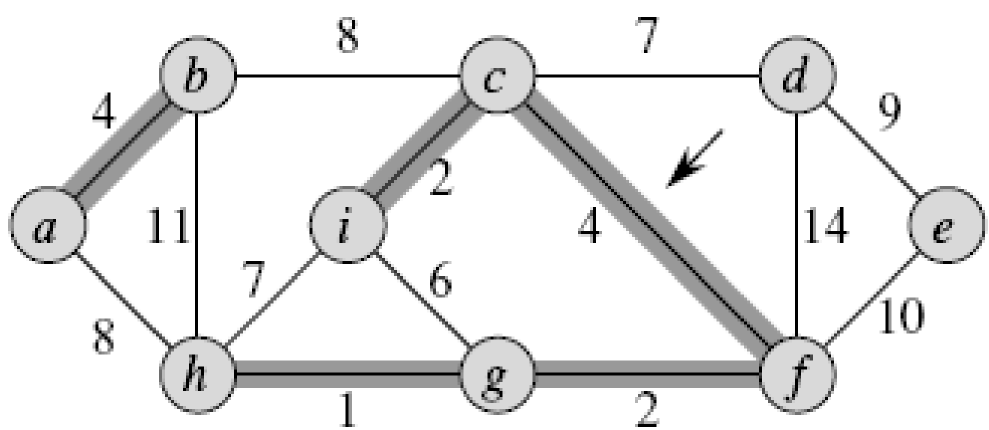
A' $(g,h);(c,i);(f,g);(a,b);(c,f);\boxed{(\cancel{g,i})};(c,d);(h,i);$
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

c e *d* pertencem a mesma árvore na floresta?

- 3º passo: para cada aresta ordenada, faç

$$\{\{a,b\},\{d\},\{e\},\{c,f,g,h,i\}\}$$



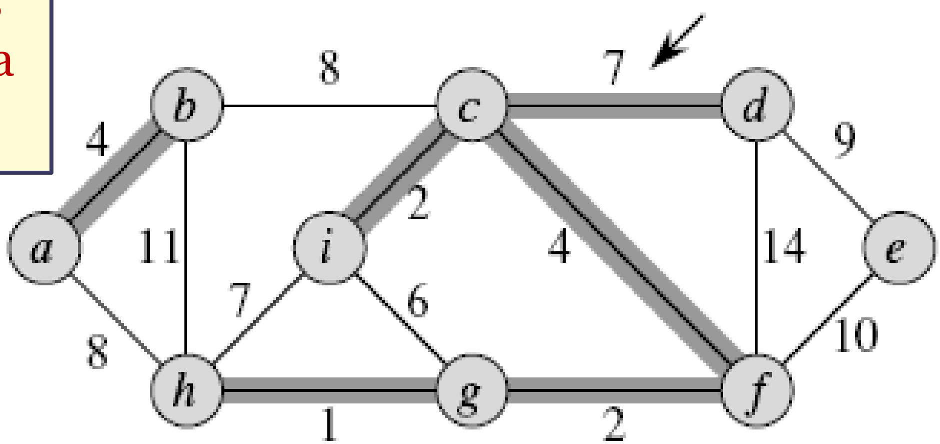
A' $(g,h);(c,i);(f,g);(a,b);(c,f);(\cancel{g,i});\boxed{(c,d)};(h,i);$
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{a,b\},\{e\},\boxed{\{c,d,f,g,h,i\}}\}$$

Não... Então... União das árvores de *c* e *d* e adição da aresta na AGM



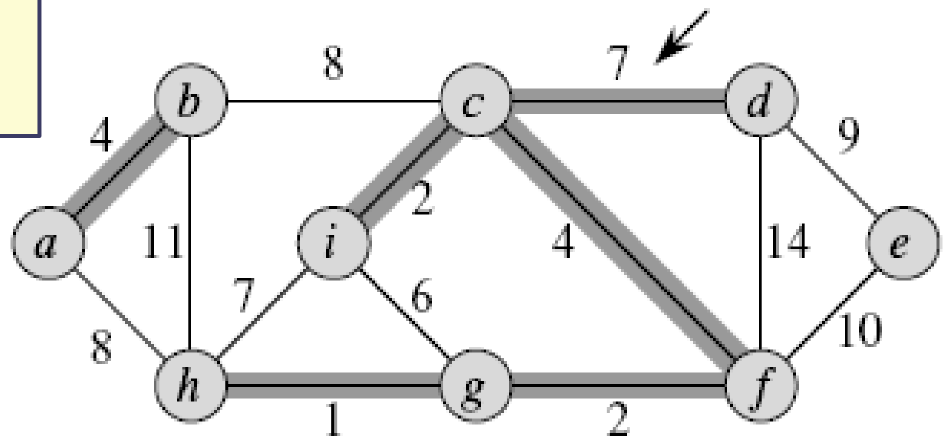
A' $(g,h); (c,i); (f,g); (a,b); (c,f); (\cancel{g,i}); \boxed{(c,d)}; (h,i);$
$(a,h); (b,c); (d,e); (e,f); (b,h); (d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

h e *i* pertencem a mesma árvore na floresta?

$$\{\{a,b\},\{e\},\{c,d,f,g,h,i\}\}$$



A' $(g,h);(c,i);(f,g);(a,b);(c,f);(\cancel{g,i});(c,d);\boxed{(h,i);}$

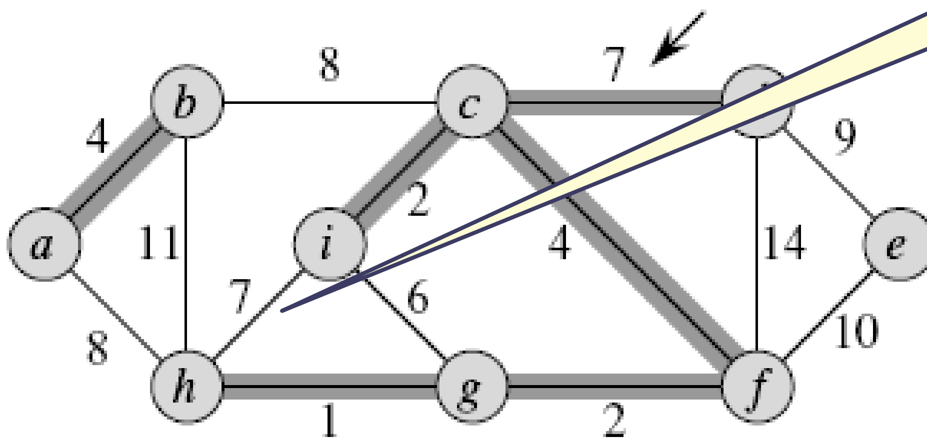
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{a,b\},\{e\},\boxed{\{c,d,f,g,h,i\}}\}$$

Fecha ciclo...



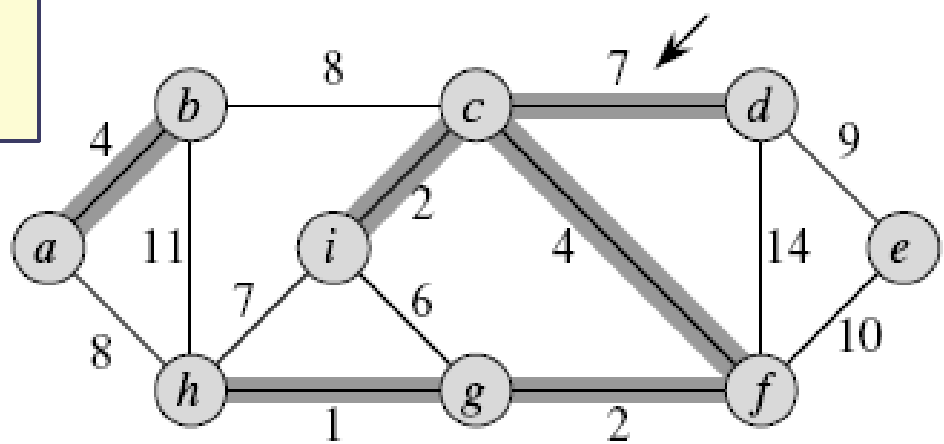
A' $(g,h);(c,i);(f,g);(a,b);(c,f);(\cancel{g,i});(c,d);\boxed{(\cancel{h,i})};$
$(a,h);(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

a e *h* pertencem a mesma árvore na floresta?

$$\{\{a,b\},\{e\},\{c,d,f,g,h,i\}\}$$



A' $(g,h);(c,i);(f,g);(a,b);(c,f);(\cancel{g,i});(c,d);(\cancel{h,i});$

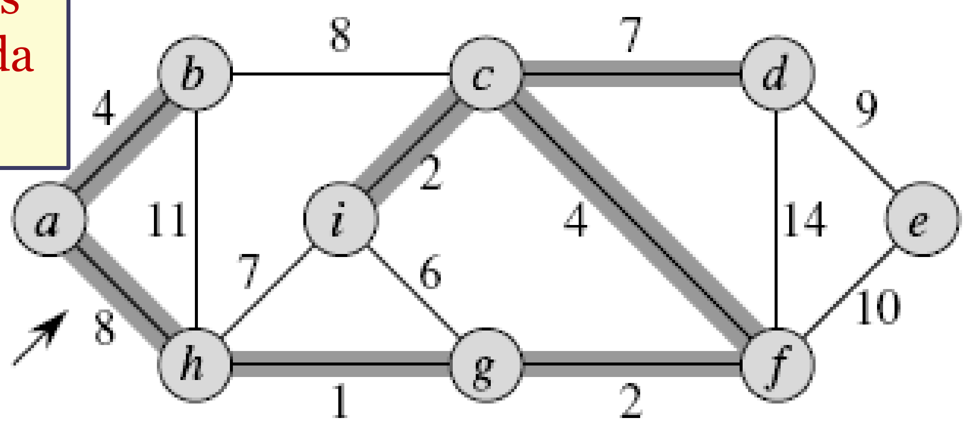
$\boxed{(a,h);}(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{e\},\boxed{\{a,b,c,d,f,g,h,i\}}\}$$

Não... Então...
União das árvores
de *a* e *h* e adição da
aresta na AGM



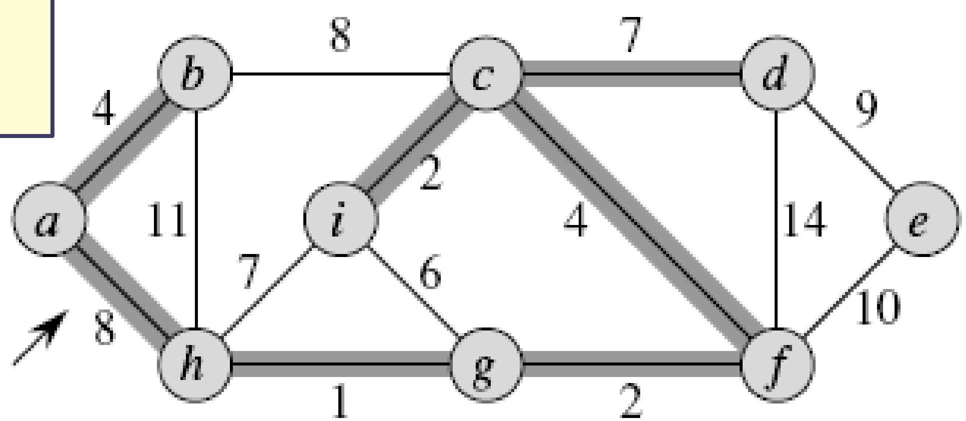
A' $(g,h);(c,i);(f,g);(a,b);(c,f);\cancel{(g,i)};(c,d);\cancel{(h,i)};$
$\boxed{(a,h)};(b,c);(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

b e *c* pertencem a mesma árvore na floresta?

$$\{\{e\},\{a,b,c,d,f,g,h,i\}\}$$

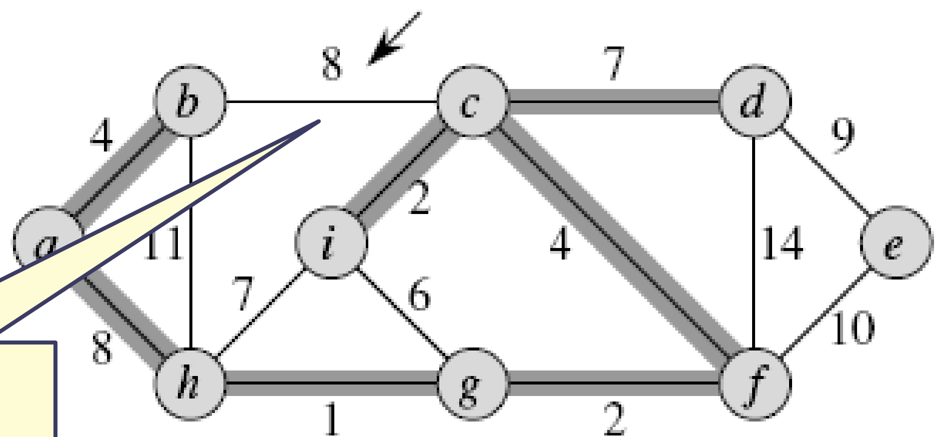


A' $(g,h);(c,i);(f,g);(a,b);(c,f);(\cancel{g,i});(c,d);(\cancel{h,i});$
$(a,h);\boxed{(b,c);}(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{e\}, \boxed{\{a,b,c,d,f,g,h,i\}}\}$$



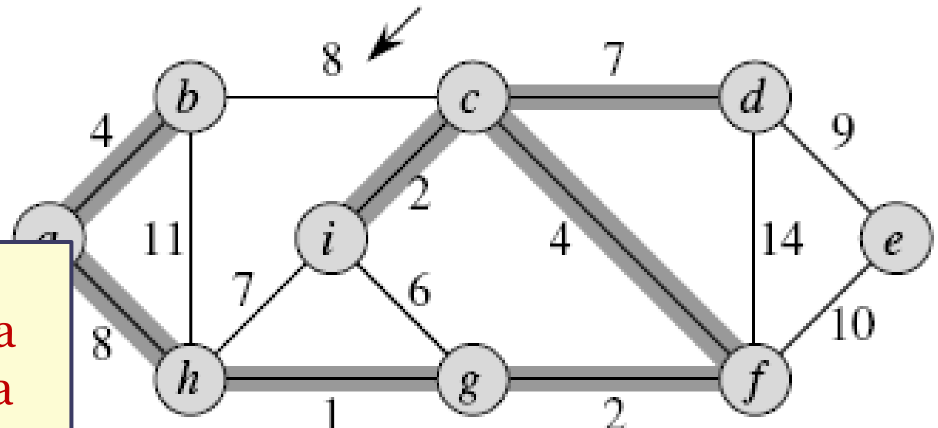
Fecha ciclo...

A' $(g,h);(c,i);(f,g);(a,b);(c,f);(\cancel{g,i});(c,d);(\cancel{h,i});$
$(a,h);\boxed{(\cancel{b,c});}(d,e);(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{e\},\{a,b,c,d,f,g,h,i\}\}$$



d e *e* pertencem a mesma árvore na floresta?

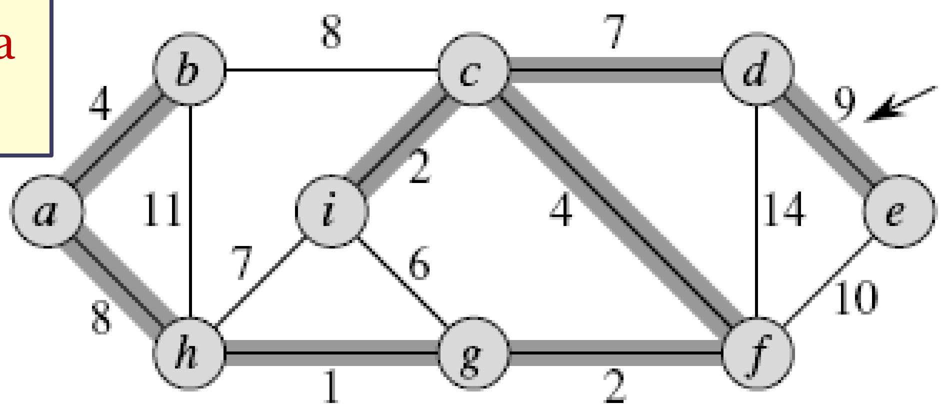
A' $(g,h);(c,i);(f,g);(a,b);(c,f);(\cancel{g,i});(c,d);(\cancel{h,i});$
$(a,h);(\cancel{b,c});\boxed{(d,e);}(e,f);(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\boxed{\{\{a, b, c, d, e, f, g, h, i\}\}}$$

Não... Então...
União das árvores
de *d* e *e* e adição da
aresta na AGM



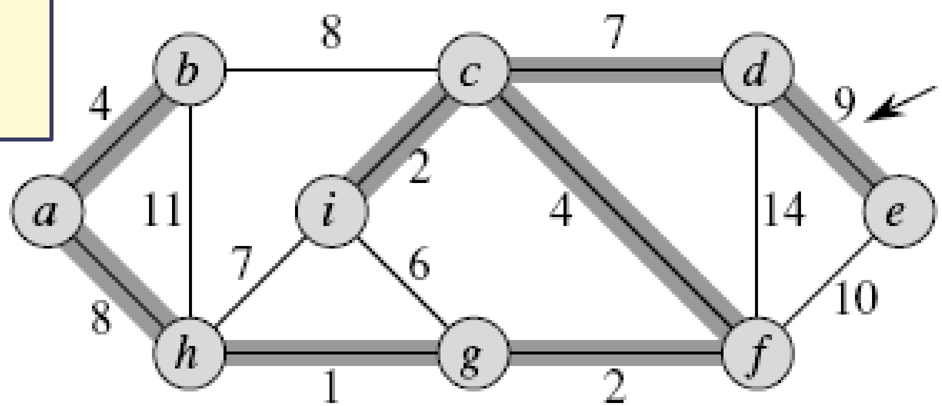
A' $(g,h); (c,i); (f,g); (a,b); (c,f); \cancel{(g,i)}; (c,d); \cancel{(h,i)};$
$(a,h); \cancel{(b,c)}; \boxed{(d,e)}; (e,f); (b,h); (d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

e e *f* pertencem a mesma árvore na floresta?

$$\{\{a,b,c,d,e,f,g,h,i\}\}$$

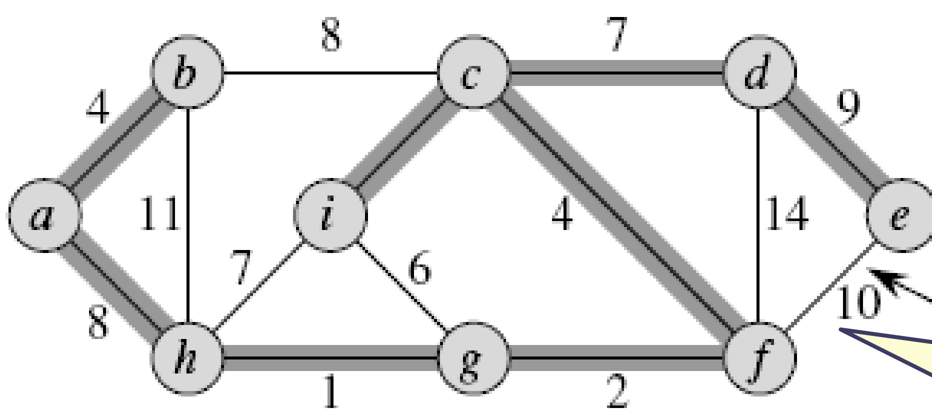


A' $(g,h);(c,i);(f,g);(a,b);(c,f);(\cancel{g,i});(c,d);(\cancel{h,i});$
$(a,h);(\cancel{b,c});(d,e);\boxed{(e,f);}(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\boxed{\{\{a,b,c,d,e,f,g,h,i\}\}}$$



Fecha ciclo...

A' $(g,h);(c,i);(f,g);(a,b);(c,f);(\cancel{g,i});(c,d);(\cancel{h,i});$

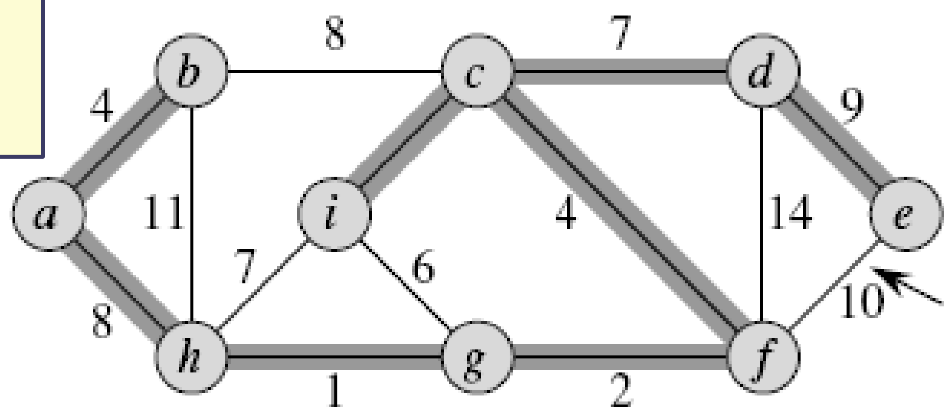
$(a,h);(\cancel{b,c});(d,e);\boxed{(\cancel{e,f})};(b,h);(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

b e *h* pertencem a mesma árvore na floresta?

$$\{\{a,b,c,d,e,f,g,h,i\}\}$$



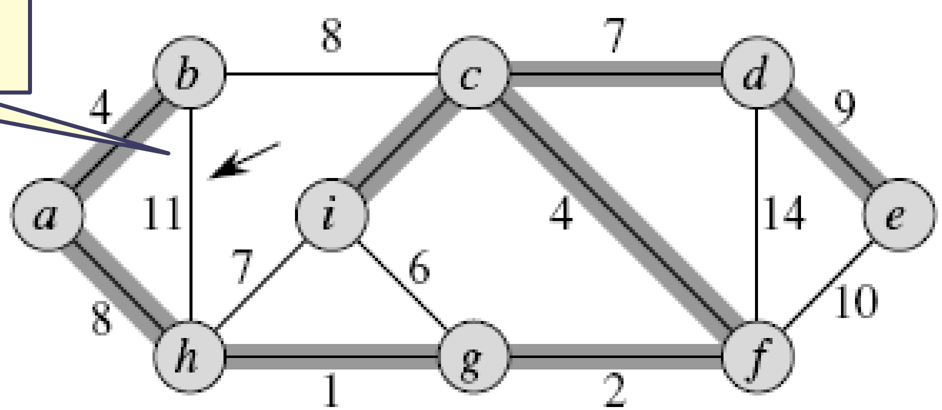
A’ $(g,h);(c,i);(f,g);(a,b);(c,f);\cancel{(g,i)};(c,d);\cancel{(h,i)};$
$(a,h);\cancel{(b,c)};(d,e);\cancel{(e,f)};\boxed{(b,h);}(d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{a, b, c, d, e, f, g, h, i\}\}$$

Fecha ciclo...



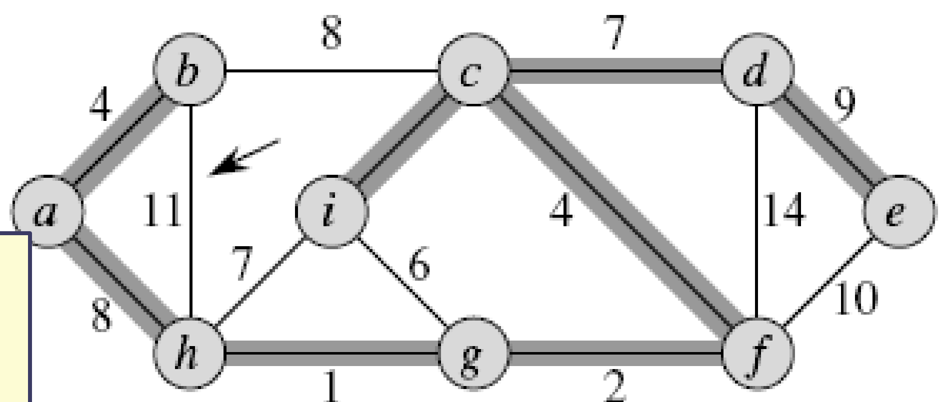
A' $(g,h); (c,i); (f,g); (a,b); (c,f); (\cancel{g,i}); (c,d); (\cancel{h,i});$

$(a,h); (\cancel{b,c}); (d,e); (\cancel{e,f}); (\cancel{b,h}); (d,f)$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{a, b, c, d, e, f, g, h, i\}\}$$



d e *f* pertencem a mesma árvore na floresta?

A' $(g,h); (c,i); (f,g); (a,b); (c,f); \cancel{(g,i)}; (c,d); \cancel{(h,i)};$

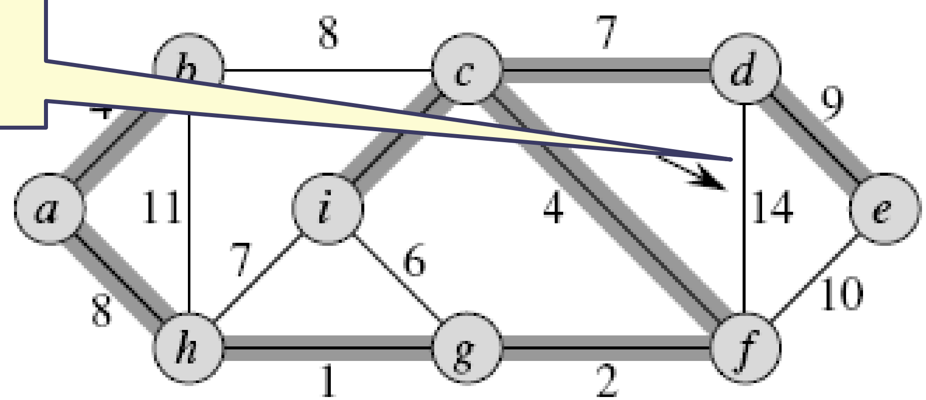
$(a,h); \cancel{(b,c)}; (d,e); \cancel{(e,f)}; \cancel{(b,h)}; \boxed{(d,f)}$

AGM utilizando Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\boxed{\{\{a,b,c,d,e,f,g,h,i\}\}}$$

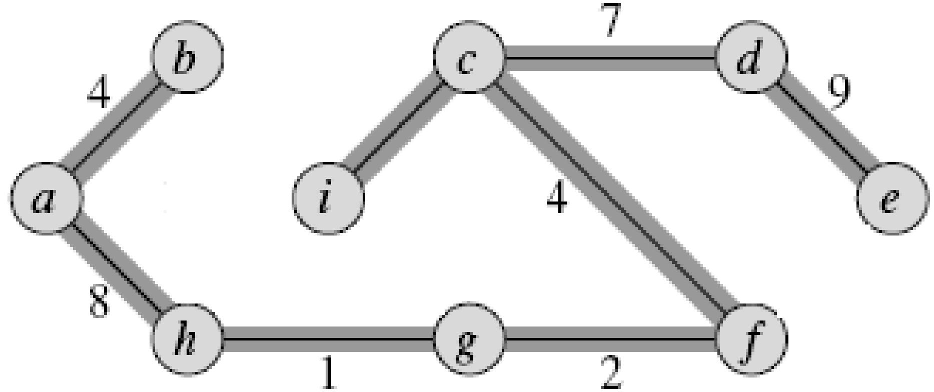
Fecha ciclo...



$$A' \quad (g,h);(c,i);(f,g);(a,b);(c,f);(\cancel{g,i});(c,d);(\cancel{h,i});$$
$$(a,h);(\cancel{b,c});(d,e);(\cancel{e,f});(\cancel{b,h});\boxed{(\cancel{d,f})}$$

AGM utilizando Kruskal

- O conjunto X (arestas da AGM) foi composto ao longo da execução do Kruskal, onde apenas as arestas não marcadas de A' foram adicionadas à árvore.



A' $(g,h); (c,i); (f,g); (a,b); (c,f); (\cancel{g,i}); (c,d); (\cancel{h,i});$
$(a,h); (\cancel{b,c}); (d,e); (\cancel{e,f}); (\cancel{b,h}); (\cancel{d,f})$

Animação na Web do algoritmo de Kruskal

- Pode ser feito passo a passo. Bom para entendimento geral do algoritmo:

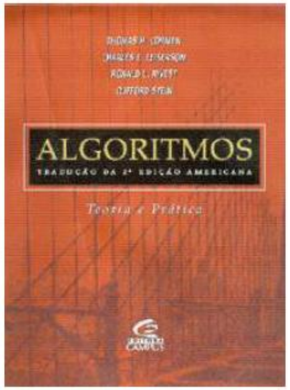
  - http://students.ceid.upatras.gr/~papagel/project/kruskal.htm

Exercício

- Qual é a complexidade do algoritmo de Kruskal?
- Ela depende quais estruturas?
- Proponha uma estrutura eficiente para armazenar e efetuar as operações na floresta e nas árvores do algoritmo de Kruskal.

Bibliografia

- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; (2002). Algoritmos – Teoria e Prática. Tradução da 2ª edição americana. Rio de Janeiro. Editora Campus.



- ZIVIANI, N. (2007). Projeto e Algoritmos com implementações em Java e C++. São Paulo. Editora Thomson;